



git



version** control **system



Bitbucket



GitLab

Popular



Local

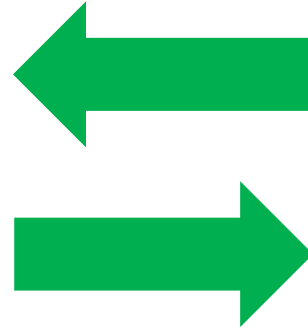


Remote Server



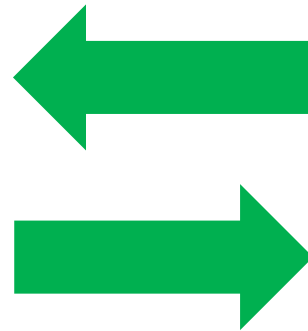
git

Repository

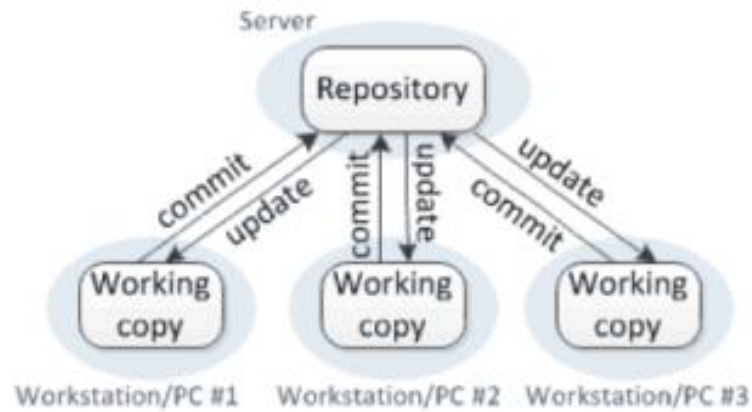


Folder

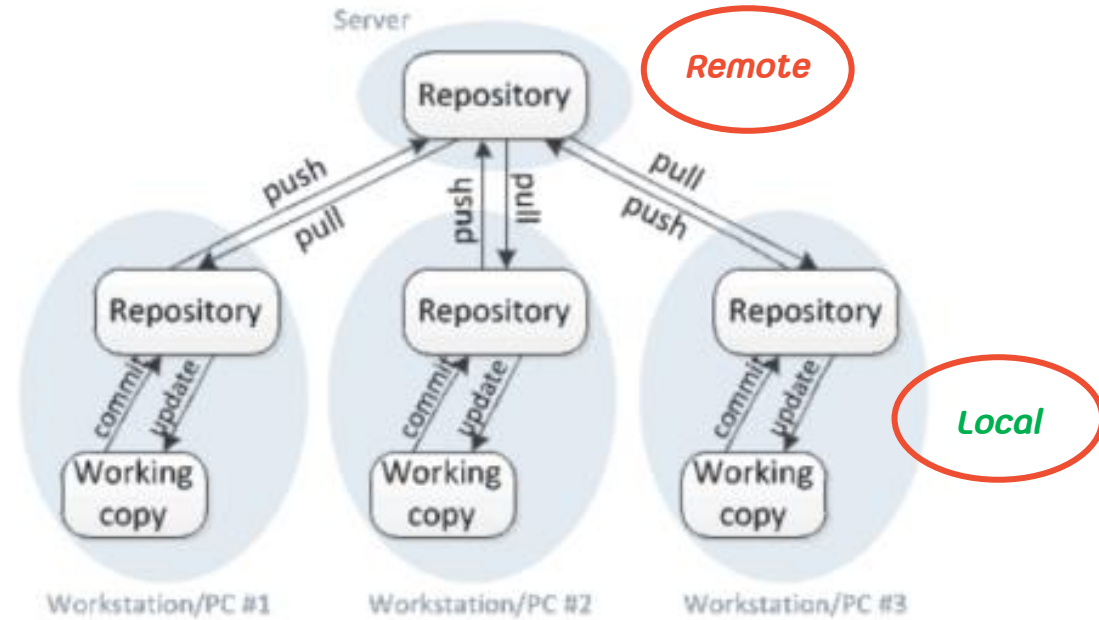
Commit



Centralized version control



Distributed version control



ต้องสมัคร Git Hub ก่อน

```
git config --global user.email "ใส่ อีเมล Github"  
git config --global user.name "User Name"
```

Git *Commands Local*

ทำตามขั้นตอนดังนี้

git init

การ **Initialize** Local Repository **สร้าง Repository** บนเครื่องคอมพิวเตอร์ของเรา สามารถใช้งาน Git ได้

git add <file name>

Add เพิ่มไฟล์เข้าไปที่ Repository

git add .

Add เพิ่มไฟล์ทุกไฟล์ที่ยังไม่ได้เพิ่มเข้า Repository

git status

ดูสถานะของไฟล์ ถ้าขึ้นสีเขียว **Changes to be committed** แสดงว่า พร้อม commit

git commit -m "เรื่อง เนื้อหาที่ Commit"

Commit file เข้าไว้ที่ **Repository** หรือการ Save

git commit -a -m "เรื่อง เนื้อหาที่ Commit"

Commit file เข้าไว้ที่ **Repository** พร้อม **add file** เข้าไปด้วย พร้อมกัน

git log

ดูประวัติของ **git** ว่ามีการกระทำอะไรบ้าง **Add** อะไรเข้ามา **commit** อะไรไว้บ้าง

Q

ออกจาก **git log**

git checkout <Hash ID>

Rollback หรือย้อนกลับไปจุดที่เคย **Commit** ไว้

Git Commands To Server

กรณีที่ยังไม่สร้าง Repository ในเครื่อง...or create a new repository on the command line

echo "# test" >> README.md สร้างไฟล์ เพื่อทดสอบการอัปขึ้น Server

git init git add README.md การ Add ไฟล์เข้า Repository เครื่องเรา Local Repository

git commit -m "first commit" ทำการ Commit ลง Local Repository

git remote add origin https://github.com/nathakrit061103jnt/test.git การเชื่อมต่อเข้า Repository ของ Server

git push -u origin master อัปไฟล์ จาก Repository เครื่องเรา ขึ้น Repository ของ Server

Up Local Repository to Server Repository

กรณีที่ สร้าง Repository ในเครื่องแล้ว อยากอัปขึ้น Server

...or push an existing repository from the command line

git remote add origin https://github.com/nathakrit061103jnt/test.git

git push -u origin master

Git Commands To Server

`git remote add origin <url Repository>.git`

การรีโมทเข้า Repository Server

`git push -u origin master` หรือ `git push` ก็ได้

การอัปไฟล์เข้า Repository Server

Git *Commands To* *Server*

Git Clone

การทำ *Git Clone* เพื่อให้คนในทีมสามารถนำโปรเจ็กต์จาก *Repository* เราไปพัฒนาร่วมกัน

git clone <url Repository>

Example

Git clone https://github.com/nathakrit061103jnt/test.git

Git Commands To Server

Push

การอัปไฟล์ขึ้น Server

คำสั่ง *git push*

Pull

การ Download หรือ Update Code ล่าสุด

คำสั่ง *git pull*

Git Commands To Server

Git Push And Git Pull

ทำตามขั้นตอนดังนี้

git add .

การทำ อัปไฟล์ขึ้น Repository Server

git commit -m "ข้อความที่ต้องการ Commit" ทำการ Commit file เข้า Local Repository

git pull

ดึงโค้ดที่อยู่บน Server ที่เป็นเวอร์ชันปัจจุบัน มาบนเครื่องเรา

git push

อัปโค้ด ที่แก้ไขแล้วขึ้น Server

Git Branch

การแตก Branch เพื่อ แบ่งงานกันพัฒนาคนละ Branch Repository เดียวกัน หรือว่าเป็นการแยกกันพัฒนาคนละ ฟีเจอร์ กับเพื่อน

การเรียกดู **branch** ที่กำลังทำงานอยู่ จะขึ้นสีเขียว และแสดง **branch** ทั้งหมดที่มี

`git branch`

การสร้าง **branch** ใหม่

`git branch <ชื่อ Branch ที่ต้องการสร้าง>`

การเปลี่ยน **branch**

`git checkout <ชื่อ branch ที่ต้องการ Switch หรือ ชื่อ branch ที่ต้องการเปลี่ยนเปลี่ยน>`

การสร้าง **branch** ใหม่ และทำการ การเปลี่ยน **branch Switch** ไปที่ **branch** ใหม่ที่เพิ่งสร้างขึ้น

`git checkout -b <ชื่อ branch ที่ต้องการ Switch หรือ ชื่อ branch ที่ต้องการเปลี่ยนเปลี่ยน>`

การลบ **branch** ที่ **Merge Branch** แล้ว

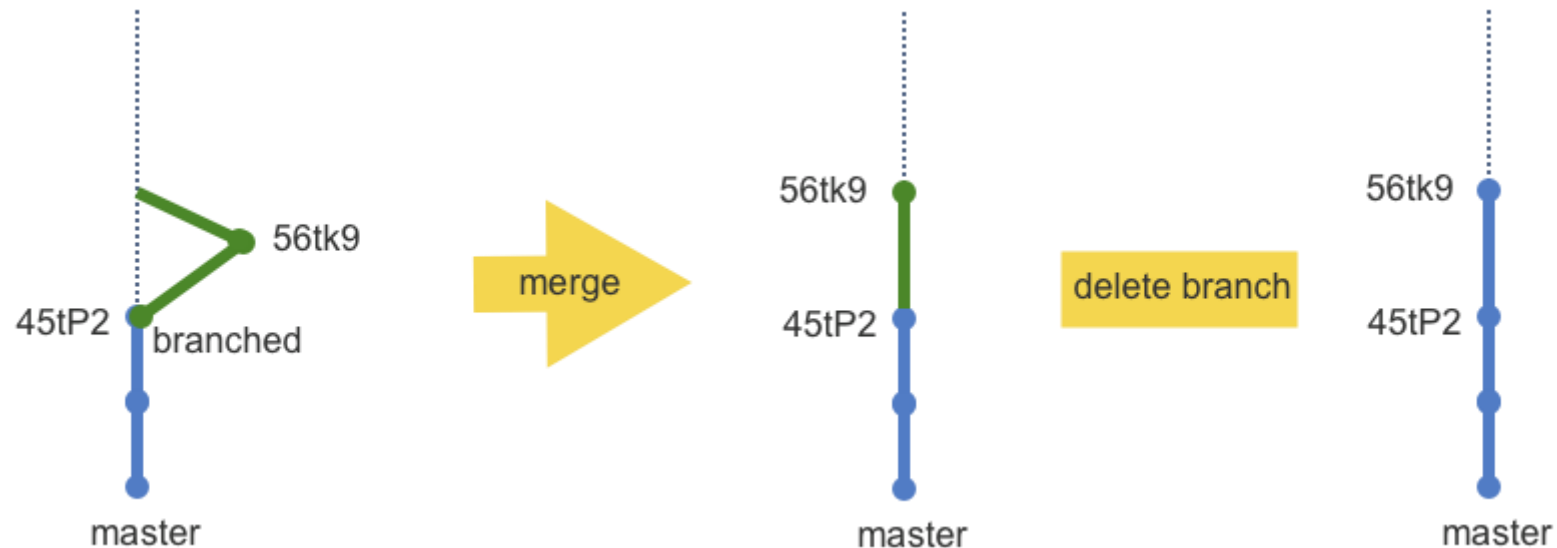
`git branch -d <ชื่อ branch ที่ต้องการลบ>`

การลบ **branch** ที่ยังไม่ **Merge Branch** แล้ว

`git branch -D <ชื่อ branch ที่ต้องการลบ>`

Merge Branch

fast forward & config



คำสั่ง ในการ Merge Branch ::> `git merge <ชื่อ branch>`

Merge Branch

ทำตามขั้นตอนดังนี้

git commit -a -m "ข้อความที่ต้องการ Commit"

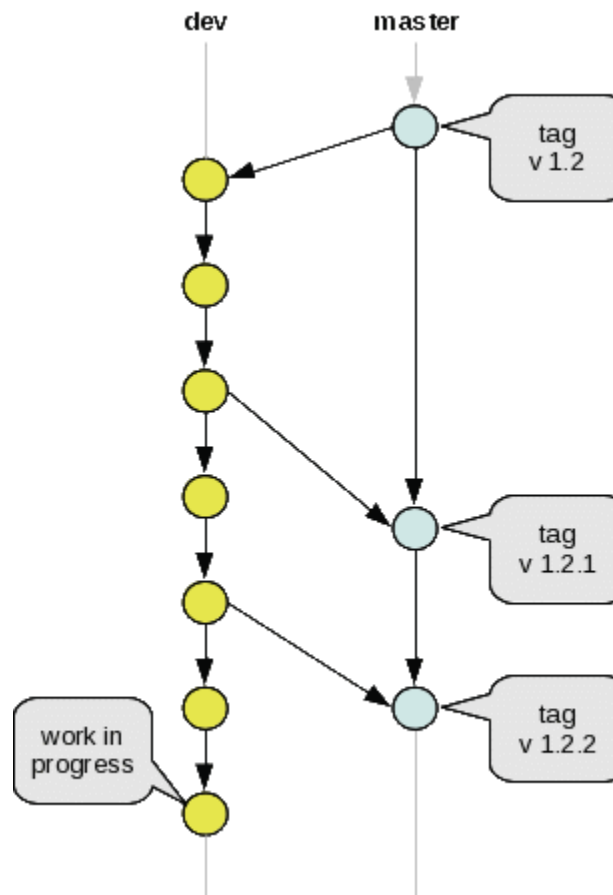
ทำการ Add file และ commit ไปพร้อมกัน

git merge <ชื่อ branch>

ทำการ Merge Branch

Git Tag

เหมือนการ Zip file เป็น Version ไว้ สามารถ Commit ได้แต่ ไม่สามารถเปลี่ยนแปลง สิ่งที่อยู่ใน Tag ได้ ซึ่งต่างจาก Branch ที่ แก้ไขได้



===== ***GIT TAG*** =====

git tag // show tag

git tag <ชื่อ tag ที่ต้องการสร้าง> //create tag

git tag -d <ชื่อ tag ที่ต้องการลบ> //ลบ tag

git tag <ชื่อ tag ที่ต้องการเปลี่ยนไปที่ > //เปลี่ยนไปที่ tag

git push origin <ชื่อ tag ที่ต้องการ อัป ขึ้น server> // อัป tag ขึ้น server

git push -d origin <ชื่อ tag ที่ต้องการลบ บน server> // ลบ tag บน server

Git Fetch

เป็นการ ดึงทุกอย่างที่อยู่บน server ที่เป็น สถานะปัจจุบัน คล้ายกับ GIT Pull ซึ่ง Git Fetch จะดึงมาทั้ง Branch และ Tag บนเซิร์ฟเวอร์

คำสั่ง

git fetch